Class 15

Pierce Ford (PID: A59010464)

11/17/2021

Background

Today we examine a published RNA-seq experiment where airway smooth muscle cells were treated with dexamethasone, a synthetic glucocorticoid steroid with anti-inflammatory effects (Himes et al. 2014).

Load the contData and colData

```
counts <- read.csv("airway_scaledcounts.csv", row.names=1)
metadata <-  read.csv("airway_metadata.csv")

#Examine counts
nrow(counts)
```

```
## [1] 38694
```

```
head(counts)
```

```
##                 SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516
## ENSG00000000003        723        486        904        445       1170
## ENSG00000000005          0          0          0          0          0
## ENSG00000000419        467        523        616        371        582
## ENSG00000000457        347        258        364        237        318
## ENSG00000000460         96         81         73         66        118
## ENSG00000000938          0          0          1          0          2
##                 SRR1039517 SRR1039520 SRR1039521
## ENSG00000000003       1097        806        604
## ENSG00000000005          0          0          0
## ENSG00000000419        781        417        509
## ENSG00000000457        447        330        324
## ENSG00000000460         94        102         74
## ENSG00000000938          0          0          0
```

```
#Examine metadata
head(metadata)
```

```
##           id     dex celltype     geo_id
## 1 SRR1039508 control   N61311 GSM1275862
```

```
## 2 SRR1039509 treated   N61311 GSM1275863
## 3 SRR1039512 control  N052611 GSM1275866
## 4 SRR1039513 treated  N052611 GSM1275867
## 5 SRR1039516 control  N080611 GSM1275870
## 6 SRR1039517 treated  N080611 GSM1275871
```

There are 38694 genes in this dataset.

How can we check correspondence of the metadata and count data setup?

```
#View the metadata row names and counts columns
metadata$id
```

```
## [1] "SRR1039508" "SRR1039509" "SRR1039512" "SRR1039513" "SRR1039516"
## [6] "SRR1039517" "SRR1039520" "SRR1039521"
```

```
colnames(counts)
```

```
## [1] "SRR1039508" "SRR1039509" "SRR1039512" "SRR1039513" "SRR1039516"
## [6] "SRR1039517" "SRR1039520" "SRR1039521"
```

```
#make sure they are the same
all(metadata$id == colnames(counts))
```

```
## [1] TRUE
```

Compare Control to Treated

Let's average the data between controls and treated samples to begin a simple analysis.

```
control.inds <- metadata$dex == "control"
control.names <- metadata[control.inds, "id"]
```

Use the control names to access the corresponding columns of the `counts` data.

```
control.data <- counts[,control.names]
control.mean <- rowMeans(control.data)
```

Repeat for treated.

```
treated.inds <- metadata$dex == "treated"
treated.names <- metadata[treated.inds, "id"]
treated.data <- counts[,treated.names]
treated.mean <- rowMeans(treated.data)
```

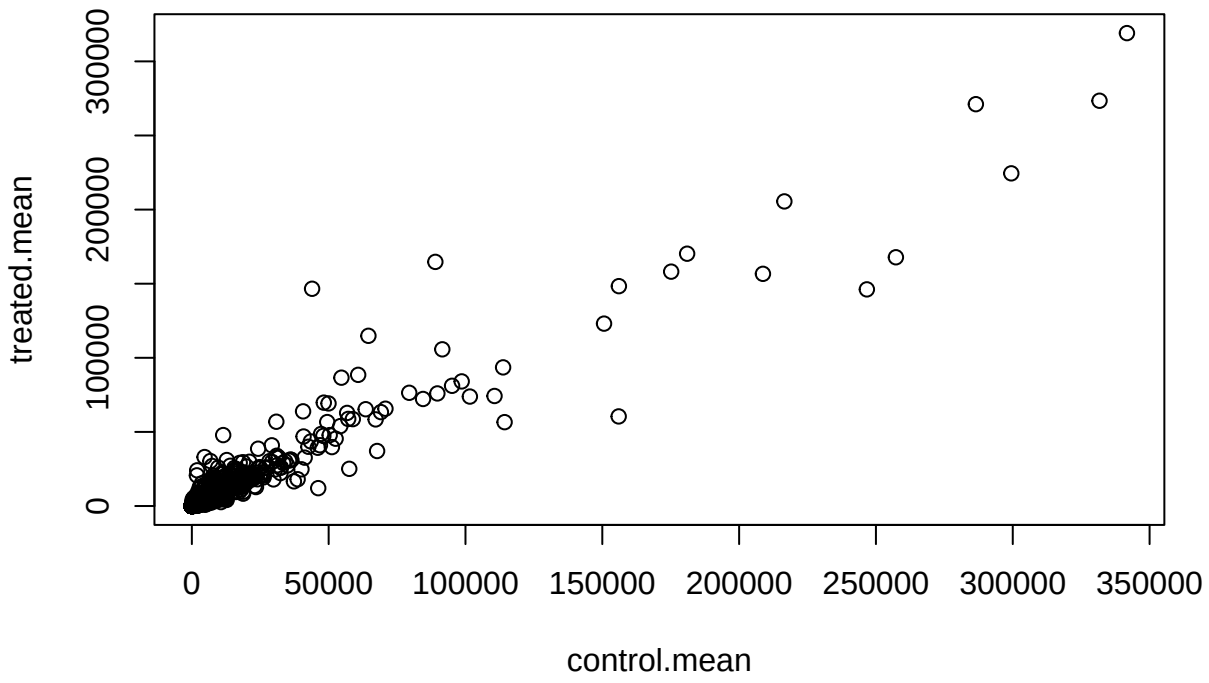
Combine the averaged data for bookkeeping.

```
meancounts <- data.frame(control.mean, treated.mean)
```

Compare the control and treated

Quick visualization with base R.

```
plot(meancounts)
```

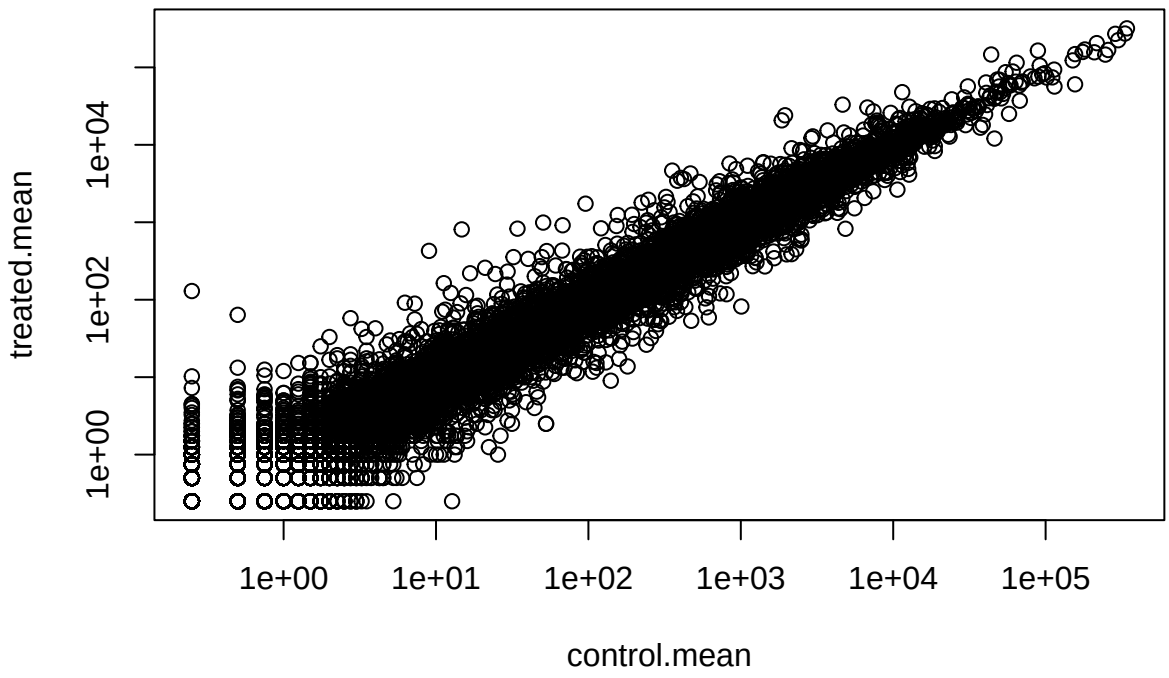


This would benefit from log transformation.

```
plot(meancounts, log="xy")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 15032 x values <= 0 omitted
## from logarithmic plot
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 15281 y values <= 0 omitted
## from logarithmic plot
```



Log transformations often make data visualization much nicer, base 2 is common.

```
meancounts$log2FC <- log2(meancounts[,"treated.mean"]/meancounts[,"control.mean"])
head(meancounts)
```

```
##                 control.mean treated.mean      log2FC
## ENSG00000000003       900.75       658.00 -0.45303916
## ENSG00000000005         0.00         0.00         NaN
## ENSG00000000419       520.50       546.00  0.06900279
## ENSG00000000457       339.75       316.50 -0.10226805
## ENSG00000000460        97.25        78.75 -0.30441833
## ENSG00000000938         0.75         0.00        -Inf
```

Remove data with zero reads in either control or treated cells.

```
zero.vals <- which(meancounts[,1:2]==0, arr.ind=TRUE)
to.rm <- unique(zero.vals[,1])
meancounts.filtered <- meancounts[-to.rm,]

#Examine filtered dataset
head(meancounts.filtered)
```

```
##                 control.mean treated.mean      log2FC
## ENSG00000000003       900.75       658.00 -0.45303916
## ENSG00000000419       520.50       546.00  0.06900279
```

```
## ENSG00000000457       339.75       316.50 -0.10226805
## ENSG00000000460        97.25        78.75 -0.30441833
## ENSG00000000971      5219.00      6687.50  0.35769358
## ENSG00000001036      2327.00      1785.75 -0.38194109
```

```
nrow(meancounts.filtered)
```

```
## [1] 21817
```

We now have 21817 remaining.

What fraction of these genes are upregulated? Downregulated?

```
#Upregulated percent
round(100*(sum(meancounts.filtered$log2FC > 2)/nrow(meancounts.filtered)),2)
```

```
## [1] 1.15
```

```
#Downregulated percent
round(100*(sum(meancounts.filtered$log2FC < -2)/nrow(meancounts.filtered)),2)
```

```
## [1] 1.68
```

DESeq2 analysis

Load DESeq.

```
library(DESeq2)
```

```
## Loading required package: S4Vectors

## Loading required package: stats4

## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## Loading required package: Biobase
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```
citation("DESeq2")
```

```
##
##   Love, M.I., Huber, W., Anders, S. Moderated estimation of fold change
##   and dispersion for RNA-seq data with DESeq2 Genome Biology 15(12):550
##   (2014)
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2},
##     author = {Michael I. Love and Wolfgang Huber and Simon Anders},
##     year = {2014},
##     journal = {Genome Biology},
##     doi = {10.1186/s13059-014-0550-8},
##     volume = {15},
##     issue = {12},
##     pages = {550},
##   }
```

First need to set up the DESeq input object.

```
dds <- DESeqDataSetFromMatrix(countData=counts,
                              colData=metadata,
                              design=~dex)
```

```
## converting counts to integer mode

## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
dds
```

```
## class: DESeqDataSet
## dim: 38694 8
## metadata(1): version
## assays(1): counts
## rownames(38694): ENSG00000000003 ENSG00000000005 ... ENSG00000283120
##   ENSG00000283123
## rowData names(0):
## colnames(8): SRR1039508 SRR1039509 ... SRR1039520 SRR1039521
## colData names(4): id dex celltype geo_id
```

Run the DESeq analysis.

```
dds <- DESeq(dds)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

Open results.

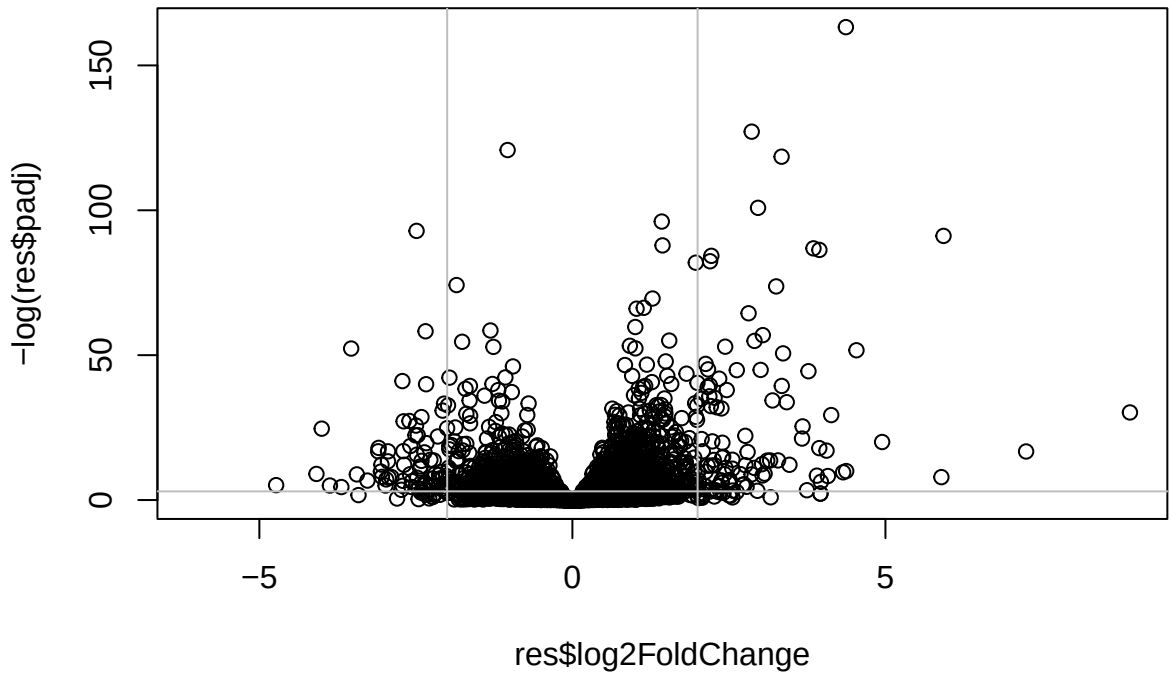
```
res <- results(dds)
head(res)
```

```
## log2 fold change (MLE): dex treated vs control
## Wald test p-value: dex treated vs control
## DataFrame with 6 rows and 6 columns
##                   baseMean log2FoldChange     lfcSE      stat    pvalue
##                  <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000000003 747.194195     -0.3507030  0.168246 -2.084470 0.0371175
## ENSG00000000005   0.000000             NA        NA        NA        NA
## ENSG00000000419 520.134160      0.2061078  0.101059  2.039475 0.0414026
## ENSG00000000457 322.664844      0.0245269  0.145145  0.168982 0.8658106
## ENSG00000000460  87.682625     -0.1471420  0.257007 -0.572521 0.5669691
## ENSG00000000938   0.319167     -1.7322890  3.493601 -0.495846 0.6200029
##                      padj
##                 <numeric>
## ENSG00000000003  0.163035
## ENSG00000000005        NA
## ENSG00000000419  0.176032
## ENSG00000000457  0.961694
## ENSG00000000460  0.815849
## ENSG00000000938        NA
```

Visualizing with a Volcano Plot

This is a really common visualization technique for this type of data.

```
plot(res$log2FoldChange, -log(res$padj))
abline(v=c(-2,2), col="gray")
abline(h=-log(0.05), col="gray")
```



Adding Annotation Data

We want to add meaningful gene names to our dataset so we can make some biological sense of it.

To do this we will use two bioconductor packages, one does the work and is called **AnnotationDbi** and the other contains the data we are going to map between and is called **org.Hs.eg.db**

```
library(AnnotationDbi)
library(org.Hs.eg.db)
```

```
##
```

We can use the mapIds function to add the gene symbol (commonly used gene name) to our dataset.

```
res$symbol <- mapIds(org.Hs.eg.db,
                     keys=row.names(res), # Our genenames
                     keytype="ENSEMBL",        # The format of our genenames
                     column="SYMBOL",          # The new format we want to add
                     multiVals="first")
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
head(res)
```

```
## log2 fold change (MLE): dex treated vs control
## Wald test p-value: dex treated vs control
## DataFrame with 6 rows and 7 columns
##                   baseMean log2FoldChange     lfcSE      stat    pvalue
##                  <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000000003 747.194195     -0.3507030  0.168246 -2.084470 0.0371175
## ENSG00000000005   0.000000             NA        NA        NA        NA
## ENSG00000000419 520.134160      0.2061078  0.101059  2.039475 0.0414026
## ENSG00000000457 322.664844      0.0245269  0.145145  0.168982 0.8658106
## ENSG00000000460  87.682625     -0.1471420  0.257007 -0.572521 0.5669691
## ENSG00000000938   0.319167     -1.7322890  3.493601 -0.495846 0.6200029
##                      padj      symbol
##                 <numeric> <character>
## ENSG00000000003  0.163035      TSPAN6
## ENSG00000000005        NA        TNMD
## ENSG00000000419  0.176032        DPM1
## ENSG00000000457  0.961694       SCYL3
## ENSG00000000460  0.815849    C1orf112
## ENSG00000000938        NA         FGR
```

Save our results to a CSV for later

```
write.csv(res, file="allmyresults.csv")
```

Pathway Analysis

Bring biology into this analysis using KEGG.

```
#Load necessary packages
library(pathview)
```

```
## ##############################################################################
## Pathview is an open source software package distributed under GNU General
## Public License version 3 (GPLv3). Details of GPLv3 is available at
## http://www.gnu.org/licenses/gpl-3.0.html. Particullary, users are required to
## formally cite the original Pathview paper (not just mention it) in publications
## or products. For details, do citation("pathview") within R.
##
```

```
## The pathview downloads and uses KEGG data. Non-academic uses may require a KEGG
## license agreement (details at http://www.kegg.jp/kegg/legal.html).
## ##############################################################################
```

```
library(gage)
```

```
##
```

```
library(gageData)

data(kegg.sets.hs)

# Examine the first 2 pathways in this kegg set for humans
head(kegg.sets.hs, 2)
```

```
## $`hsa00232 Caffeine metabolism`
## [1] "10"   "1544" "1548" "1549" "1553" "7498" "9"
##
## $`hsa00983 Drug metabolism - other enzymes`
##  [1] "10"     "1066"   "10720"  "10941"  "151531" "1548"   "1549"   "1551"
##  [9] "1553"   "1576"   "1577"   "1806"   "1807"   "1890"   "221223" "2990"
## [17] "3251"   "3614"   "3615"   "3704"   "51733"  "54490"  "54575"  "54576"
## [25] "54577"  "54578"  "54579"  "54600"  "54657"  "54658"  "54659"  "54963"
## [33] "574537" "64816"  "7083"   "7084"   "7172"   "7363"   "7364"   "7365"
## [41] "7366"   "7367"   "7371"   "7372"   "7378"   "7498"   "79799"  "83549"
## [49] "8824"   "8833"   "9"      "978"
```

In order to map our data to KEGG pathways, we need to add gene identifiers in the ENTREZ format.

```
res$entrez <- mapIds(org.Hs.eg.db,
                     keys=row.names(res), # Our genenames
                     keytype="ENSEMBL",        # The format of our genenames
                     column="ENTREZID",          # The new format we want to add
                     multiVals="first")
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
res$genename<- mapIds(org.Hs.eg.db,
                     keys=row.names(res), # Our genenames
                     keytype="ENSEMBL",        # The format of our genenames
                     column="GENENAME",          # The new format we want to add
                     multiVals="first")
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
#Check that the new identifiers were added
head(res)
```

```
## log2 fold change (MLE): dex treated vs control
## Wald test p-value: dex treated vs control
```

```
## DataFrame with 6 rows and 9 columns
##                    baseMean log2FoldChange     lfcSE      stat    pvalue
##                   <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000000003 747.194195     -0.3507030  0.168246 -2.084470 0.0371175
## ENSG00000000005   0.000000             NA        NA        NA        NA
## ENSG00000000419 520.134160      0.2061078  0.101059  2.039475 0.0414026
## ENSG00000000457 322.664844      0.0245269  0.145145  0.168982 0.8658106
## ENSG00000000460  87.682625     -0.1471420  0.257007 -0.572521 0.5669691
## ENSG00000000938   0.319167     -1.7322890  3.493601 -0.495846 0.6200029
##                       padj      symbol      entrez               genename
##                  <numeric> <character> <character>            <character>
## ENSG00000000003  0.163035      TSPAN6        7105          tetraspanin 6
## ENSG00000000005        NA        TNMD       64102            tenomodulin
## ENSG00000000419  0.176032        DPM1        8813 dolichyl-phosphate m..
## ENSG00000000457  0.961694       SCYL3       57147 SCY1 like pseudokina..
## ENSG00000000460  0.815849    C1orf112       55732 chromosome 1 open re..
## ENSG00000000938        NA         FGR        2268 FGR proto-oncogene, ..
```

The main gage() function requires a named vector of fold changes, where the names of the values are the Entrez gene IDs.

Note that we used the mapIDs() function above to obtain Entrez gene IDs (stored in res$entrez) and we have the fold change results from DESeq2 analysis (stored in res$log2FoldChange).

```
#Create the vector
foldchanges <- res$log2FoldChange

#Give it names
names(foldchanges) <- res$entrez

#Confirm it worked
head(foldchanges)
```

```
##        7105       64102        8813       57147       55732        2268
## -0.35070302          NA  0.20610777  0.02452695 -0.14714205 -1.73228897
```

Now we can use `gage()`.

```
#Get results
keggres = gage(foldchanges, gsets=kegg.sets.hs)

#View attributes
attributes(keggres)
```

```
## $names
## [1] "greater" "less"    "stats"
```

```
#View keggres
head(keggres$greater)
```

```
##                                        p.geomean stat.mean       p.val
## hsa00500 Starch and sucrose metabolism  0.002822007  2.825461 0.002822007
```

```
## hsa00330 Arginine and proline metabolism 0.012317455  2.280002 0.012317455
## hsa04910 Insulin signaling pathway       0.017110962  2.129511 0.017110962
## hsa04510 Focal adhesion                  0.025239833  1.961955 0.025239833
## hsa04920 Adipocytokine signaling pathway 0.043426078  1.725063 0.043426078
## hsa00790 Folate biosynthesis             0.048254489  1.744387 0.048254489
##                                              q.val set.size        exp1
## hsa00500 Starch and sucrose metabolism   0.6010875       54 0.002822007
## hsa00330 Arginine and proline metabolism 0.7774866       54 0.012317455
## hsa04910 Insulin signaling pathway       0.7774866      138 0.017110962
## hsa04510 Focal adhesion                  0.7774866      200 0.025239833
## hsa04920 Adipocytokine signaling pathway 0.7774866       68 0.043426078
## hsa00790 Folate biosynthesis             0.7774866       11 0.048254489
```

```
head(keggres$less)
```

```
##                                                      p.geomean stat.mean
## hsa05332 Graft-versus-host disease                  0.0004250461 -3.473346
## hsa04940 Type I diabetes mellitus                   0.0017820293 -3.002352
## hsa05310 Asthma                                     0.0020045888 -3.009050
## hsa04672 Intestinal immune network for IgA production 0.0060434515 -2.560547
## hsa05330 Allograft rejection                        0.0073678825 -2.501419
## hsa04340 Hedgehog signaling pathway                 0.0133239547 -2.248547
##                                                            p.val      q.val
## hsa05332 Graft-versus-host disease                  0.0004250461 0.09053483
## hsa04940 Type I diabetes mellitus                   0.0017820293 0.14232581
## hsa05310 Asthma                                     0.0020045888 0.14232581
## hsa04672 Intestinal immune network for IgA production 0.0060434515 0.31387180
## hsa05330 Allograft rejection                        0.0073678825 0.31387180
## hsa04340 Hedgehog signaling pathway                 0.0133239547 0.47300039
##                                                    set.size         exp1
## hsa05332 Graft-versus-host disease                       40 0.0004250461
## hsa04940 Type I diabetes mellitus                        42 0.0017820293
## hsa05310 Asthma                                          29 0.0020045888
## hsa04672 Intestinal immune network for IgA production    47 0.0060434515
## hsa05330 Allograft rejection                             36 0.0073678825
## hsa04340 Hedgehog signaling pathway                      56 0.0133239547
```

```
head(keggres$stats)
```

```
##                                       stat.mean     exp1
## hsa00500 Starch and sucrose metabolism    2.825461 2.825461
## hsa00330 Arginine and proline metabolism  2.280002 2.280002
## hsa04910 Insulin signaling pathway        2.129511 2.129511
## hsa04510 Focal adhesion                   1.961955 1.961955
## hsa04920 Adipocytokine signaling pathway  1.725063 1.725063
## hsa00790 Folate biosynthesis              1.744387 1.744387
```

`pathview()` will add our genes to a kegg pathway as colored entries.

```
pathview(gene.data=foldchanges, pathway.id="hsa05310")
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

```
## Info: Working in directory /Users/pierceford/Desktop/BGGN213/github/bggn213/class15

## Info: Writing image file hsa05310.pathview.png
```